(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 331 548 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
30.07.2003 Bulletin 2003/31

(51) Int Cl.7: G06F 3/06

(21) Application number: 03250314.6

(22) Date of filing: 17.01.2003

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT SE SI SK TR
Designated Extension States:
AL LT LV MK RO

(30) Priority: 18.01.2002 US 54451

(71) Applicant: LEXAR MEDIA, INC.
Fremont, CA 94538 (US)

(72) Inventor: Estakhri, Petro
Pleasanton, CA 94566 (US)

(74) Representative:
Beresford, Keith Denis Lewis et al
BERESFORD & Co.
2-5 Warwick Court,
High Holborn
London WC1R 5DH (GB)

(54) File management of one-time-programmable non volatile memory devices

(57) An embodiment of the present invention includes a digital equipment system comprising a host for sending commands to read or write files having sectors of information, each sector having and being modifiable on a bit-by-bit, byte-by-byte or word-by-word basis. The host being operative to receive responses to the commands. The digital equipment further including a controller device responsive to the commands and including one-time-programmable nonvolatile memory for storing information organized into sectors, based on commands received from the host, and upon commands from the host to re-write a sector, the controller device for re-writing said sector on a bit-by-bit, byte-by-byte or word-for-word basis.

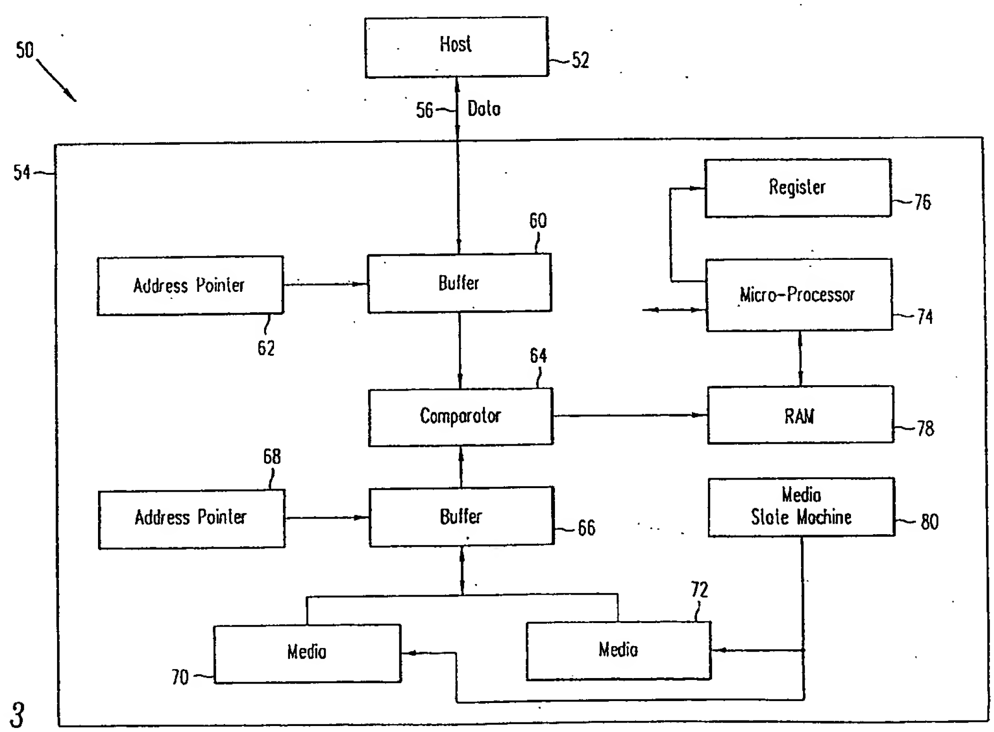

FIG. 3

EP 1 331 548 A2

Description

[0001] The present invention relates generally to methods and apparatus for managing files stored within and read from one-time-programmable nonvolatile memory devices and particularly to manipulating file managers for keeping track of the location of files stored within such one-time-programmable nonvolatile memory devices.

[0002] Storage of digital information is continuously enjoying improvement and advancements in terms its rate of performance for storage and retrieval as technology advances. As an example, digital cameras, which have become commonplace in the field of photography employ nonvolatile memory devices (devices requiring erasure of information prior to re-writing of information) for storing and retrieving captured photos of images. Similarly, archival of information is an application for nonvolatile memory devices. More recently, one-time-programmable nonvolatile memory devices have found their way for use by digital cameras and archives.

[0003] The problem posed by one-time-programmable nonvolatile memory devices is clearly the inability to re-program them. That is, once an area of memory has been written thereto, it cannot be re-written. This is precisely the reason for their natural use in archival of information. Generally, archived information requires a one-time writing or programming. The same applies to the application of one-time-programmable nonvolatile memory devices to disposable digital films. Upon storage of a number of photos of images, the disposable film is developed and disposed thereof.

[0004] The management of information, i.e. digital data, within any of these systems need be performed to maintain track of the location of the stored information. Generally, these systems include a controller device coupled to the one-time-programmable memory and to a host for directing the storage of information to particular areas within the one-time-programmable nonvolatile memory. The actual or user data is generally stored in a location identified for this purpose typically referred to as the file area and information regarding the location of the file area is kept in a separate location referred to as the system area. The file and system areas are both stored in one-time programmable nonvolatile memory. The problem that arises is successful manipulation of the system area to manage the data or file area. For example, when power is temporarily disconnected and re-established, the controller must know where there is free or available memory to write new information within the one-time-programmable nonvolatile memory and thus must restore information regarding the location of information within the latter.

[0005] Thus, the need arises for a system and method for managing or manipulating digital information stored and read from one-time-programmable nonvolatile memory.

SUMMARY OF THE INVENTION

[0006] Briefly, an embodiment of the present invention includes a digital equipment system comprising a host for sending commands to read or write files having sectors of information, each sector having and being modifyable on a bit-by-bit, byte-by-byte or word-by-word basis. The host is operative to receive responses to the commands. The digital equipment further includes a controller device responsive to the commands and includes one-time-programmable nonvolatile memory for storing information organized into sectors, based on commands received from the host, and upon commands from the host to re-write a sector, the controller device for re-writing said sector on a bit-by-bit, byte-by-byte or word-for-word basis.

[0007] A further aspect of the invention is constituted by a memory system comprising:

an interface for receiving commands from a host for the writing of data to and reading of data from the memory system;

a one-time programmable memory having a data area divided into physical sectors;

a control device responsive to the commands and comprising means for emulating a re-programmable memory whereby logical addresses available to the host are presented to the host as being mapped to re-writeable virtual sectors in the emulated memory and are actually mapped to physical addresses in the physical sectors in the one-time-programmable memory;

the control device comprising a comparator responsive to a command to rewrite a virtual sector with new data by comparing the new data with contents of a physical sector in which existing data for the virtual sector was previously programmed to identify portions of said new data which are different from the existing data; and

partial programming means for programming only the identified portions of the new data into previously unprogrammed areas of the memory and for maintaining mapping of the logical addresses for the re-written virtual sector to locations in the memory containing the new data.

[0008] The foregoing and other objects, features and advantages of the present invention will be apparent from the following detailed description of the preferred embodiments which make reference to several figures of the drawing.

IN THE DRAWINGS

[0009]

Fig. 1 shows an example of a layout 10 of a file system in accordance with an embodiment of the

present invention.

Fig. 2 shows a shows an example of a FAT entry 30.

Fig. 3 illustrates a digital equipment system 50 in accordance with an embodiment of the present invention.

Fig. 4 shows an example of the structure of data including defects pursuant to an embodiment of the present invention.

Fig. 5 shows an example of a block structure 80 in accordance with an embodiment of the present invention.

Fig. 6 illustrates an example of another embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0010] Referring now to Fig. 1, an example of a layout 10 of a file system is shown to include a system area 12 and a data area 14 in accordance with an embodiment of the present invention. The system area 12 is for storing information pertaining to the organization of the information stored or to be stored in the data area 14.

[0011] Within the system area 12 is included an area 16 for maintaining the Original Engineering Manufacturer (OEM) identification/Bidirectional Input/Output System (BIOS), an area 18 for maintaining a File Allocation Table (FAT) 1, an area 20 for maintaining a FAT 2, an area 22 for maintaining root disk directory information and an area 24 for maintaining the file area. All of the areas 16-24 are stored within a one-time-programmable memory device. As previously described, such a memory can be programmed only once in any of its given locations and thus cannot be erased and reprogrammed again. The challenged posed, which is resolved by the present invention, is to manipulate the information or data stored within the area 24 by using the information stored within and read from the system area 12.

[0012] In one embodiment of the present invention, it is further important to maintain the layout 10 of Fig. 1 in order to ensure backward compatibility of the methods and teachings of the present invention with that of prior art systems, as file system structures generally employ such a layout. In fact, one type of file system, such as a DOS-compatible file system, must follow that of the layout 10 of Fig. 1. Digital equipment, such as digital cameras, use DOS structure file systems to store digital photos (or files) into storage devices.

[0013] In Fig. 1, the FAT1 and 2 each indicate how the file area 24 is allocated. The root disk directory includes a fixed number of directory entries. The file area 24 includes data files (in the case of digital cameras, this is the area in which digital photos are maintained) and subdirectory files and free data sectors available for storage therein. These areas are divided into a block of sectors accessible by a host, discussed in further detail below. In one embodiment of the present invention, these areas are typically of fixed size depending on the capacity of the media. The data in the file area 24 are organized in clusters with each cluster including a predetermined number of sectors, such as 1, 2, 4, 8, 16 or any other integer number of sectors.

[0014] The FAT 1 and 2 each include one entry per cluster. Within each entry in the FAT, there is stored a link value pointing to the next location within the cluster in which data is located. An example of this is found in Fig. 2. Fig. 2 shows an example of a FAT entry 30 within one of its sectors. Each of the locations 32 - 48 points to a cluster within the file area 24 of Fig. 1. For example, at location 36, which is the beginning or start-of-file for "File1.txt", the linked value is '0003', which is the location within the FAT entry 30 where the next location of the data within the file area 24 for the file File1.txt is located. At location '0003' or location 38, the link value is '0004' pointing to the next location and so on. The end-of-file for File1.txt is at location 40 where the value "FFFF" appears denoting the end-of-file although any other value can be used to denote the end of file. The beginning or start of the "File2.txt" is at location 42 where the link value is '0006' pointing to the next location within the entry 30 and the '0006' points to '0007', which ultimately points to an end-of-file value for "File2.txt". The value '0000' indicates a free or available cluster.

[0015] In one embodiment of the present invention, the identification of the end-of-file is important during power-up. That is, after a power interruption and upon power-up, the system must know where it can store new information such as a digital photograph or file. In doing so, the end-of-file is identified by a flag or a predetermined value, as 'FFFF' and the location following the location in which the end-of-file resides is identified as the location for the start-of-file of a new file to be stored.

[0016] In prior art digital cameras used today, since one-time-programmable nonvolatile memory devices are not utilized, rather, nonvolatile memory devices that are erasable for re-programming are utilized, the size of each file (or picture) is a different size and the media or memory can be easily modified to account for the various picture sizes. Pictures are stored onto the media and the FAT areas are updated at times when pictures (or files) are stored onto the system. Accordingly, a sector, which is a group of storage locations, within a FAT1 or FAT2 is modified more than once when one or more pictures are store onto the media. It should be noted that a sector includes more than one bit and/or byte of information and when a picture is stored, while all of the bits or bytes of the sector may not be modified, prior art methods and techniques nevertheless modify an entire sector of information within memory or the media.

[0017] In the present invention however, due to the use of one-time-programmable nonvolatile memory devices, a sector cannot be erased and reprogrammed every time a host updates or modifies the sector, i.e. a picture is stored. Thus, the present invention must avoid reprogramming of a sector every time the host updates the same.

[0018] This is accomplished by the methods and teachings of the present invention in one of two distinct methods. First, in order for the media to be backward compatible with the present file system, a controller device is utilized to accomplish this task by keeping a correlation between logical addresses and physical addresses and also by maintaining track of defective blocks, a block including one or more sectors of information. Logical addresses are host-generated addresses for identifying sectors to be accessed by the controller. Physical addresses are addresses identifying sectors within the one-time-programmable nonvolatile memory device.

[0019] Use of one-time-programmable devices entails a relationship between logical addresses and physical addresses that is one-to-one except for blocks/sectors that are deemed defective. In one embodiment of the present invention, a bit, byte or word, where a byte and word are each one or more bits, is programmed rather than an entire sector when a picture (or file) is stored. Thus, the controller maintains track of all of the sectors that are requested to be written by the host and if the host modifies or updates the file manager sectors, such as FAT1, FAT2 and/or the root directory sectors in the case of a file management system such as DOS, as depicted in Fig. 1 herein, the controller compares the sectors to be programmed (or written) to that which is actually stored in the media and identifies the byte which is modified within the sector to be written and only programs those bits, bytes or words in the media which are modified and reside within the same physical location. This is perhaps best understood with reference to a block diagram.

[0020] Referring now to Fig. 3, a digital equipment system 50 is shown to include a host 52 coupled to a controller device 54 through a data bus 56 in accordance with an embodiment of the present invention. The host 52 commands the controller device 54, via the data bus 56, to store pictures or files within a media or one-time-programmable nonvolatile memory and later may request to read such information again through the data bus 56.

[0021] The controller device 54, in Fig. 3, is shown to include a buffer 60, an address pointer 62, a comparator 64, a buffer 66, an address pointer 68, a media device 70 and a media device 72, a microprocessor 74, a register 76, a Random Access Memory (RAM) device 78 and a media state machine 80. The buffer 60 is shown coupled to the comparator 64. Also shown coupled to the comparator 64 is the buffer 66. The address pointer 62 is shown coupled to the buffer 60 and the address pointer 68 is shown coupled to the buffer 66. The buffer is also shown coupled to the media devices 70 and 72. While not shown explicitly in Fig. 3 in the interest of avoiding confusion, the microprocessor 74 executes a program for controlling the blocks depicted within the controller device 54. The program the microprocessor 74 executes resides within the RAM device 78. The media state machine 80, in part, controls the addressing of the media devices 70 and 72. The buffer 66 directs information to be stored and read from the media devices 70 and 72. The latter are one-time-programmable nonvolatile memory devices for storing pictures and/or files depending on the application of the system 50.

[0022] In operation, the buffer 60 receives commands from the host 52 to read or program information into the media devices 70 and 72. Each time such an access is commanded, by the host, to be performed by the controller device 54, certain sectors are identified for such access using the logical addressing discussed hereinabove. Using this sector information or a derivation thereof, which is at some point in time stored in the buffer 60 of Fig. 3, the comparator compares a sector to be modified or accessed by the host with those sectors to which information has been previously written. This is performed by a comparison operation by the comparator 64 between the contents of the buffer 60 and the contents of the buffer 66. In fact, this comparison operation is performed on a bit-by-bit, byte-by-byte or word-by-word basis and only those bits, bytes or words, which need to be modified due to the soon-to-be stored information, are modified. This is referred to as partial programming since only a bit, byte or word is modified rather than an entire sector. The buffer 66 receives its information from the media devices 70 and 72.

[0023] It is also possible to keep certain number of spare locations for the system area 12 of Fig. 1 particularly if the partial programming is limited. In this case, these sectors can be mapped to the spare sectors by the controller. Use of spare sectors allows for actual address mapping rather than a one-to-one correspondence between logical and physical addresses as noted above. Also, to allow for backward compatibility with DOS-like file systems, files that are of variable length need be accommodated.

[0024] In other forms of data storage using solid state or nonvolatile one-time-programmable devices where a new file manager can be incorporated, the controller/host system nevertheless need to keep track of the addressing and defect blocks and the file locations, such as start and end-of-file and the start of the next file. In one scenario, the defect management can be accomplished in byte or word up to sector or block resolution. If the resolution is small as a byte or word, the controller or host maps the data in a form of a table where those locations that are defective are identified and saved so that they can be skipped over during programming. An example of the structure of data including defects is shown in Fig. 4.

[0025] When defect handling is performed, the controller or host map the defective locations into another location that is free of defects. The start and end-of-file are identified in a one-time-programmable media since data is in contiguous format, such as block 0, 1, 2, 3, 4, ... and saved accordingly. The start and end-of-file are saved and the defective block is skipped by using a sim-

ple identifier to identify the defective block, such as a flag or a predetermined value or an address. Fig. 5 shows an example of a block structure 80 including a start-of-file location 84 and an end-of-file location 86 and a defective sector location 82, the latter of which is skipped over when writing to sectors 0 - 6. The information that was to be written to the defective sector is instead written to a location in a spare area in the system area.

[0026] In the case of a one-to-one address relationship, a defective sector/block can point to a spare location where it is replaced by the spare location. This requires that the defective sector/block have enough usable bytes where the location of the redirected sector can be saved. For the purpose of the integrity of the saved address, the controller should employ a unique method to guarantee the value saved in the spare location. This method can be programming the address in the defective sector in multiple locations and when accessed, the locations will be compared to validate the correct address.

[0027] An example 100 of another embodiment of the present invention will be explained with reference to Fig. 6. In the event of a power failure, some of the data stored within certain locations of the one-time programmable nonvolatile memory may be corrupted. Upon power restoration, the system needs to recognize the occurrence of a power failure so that a file that includes corrupted data will be identified as an invalid file and following files will be stored at a valid location not adversely affected by power failures.

[0028] In order to recognize that a particular sector is corrupt, the system verifies the error correction code (ECC) associated with the particular sector, if the ECC indicates that the particular sector or any portions thereof is corrupt, the next sector is read.

[0029] Additionally, the system continues to check for an end-of-file identifier to determine whether or not the file has been properly written and that the ECC error is due to grown error(s) rather than power failure related errors. If there is no start-of-file identifier at a location following the corrupt sector or there is no end-of-file in the rest of the media, such location can be identified as a corrupt sector due to power failure and designated accordingly so as to prevent future storage of information therein.

[0030] If the particular sector has been determined to be corrupted, the next sector is read and if all of the data in the next sector is in a non-programmable state, i.e. all 0's or all 1's, and/or the ECC associated with the next sector indicates error, then it is determined that the end-of-file has been reached. Alternatively, to check for end-of-file more thoroughly, the sector following the next sector is read and the same criteria is implemented, i.e. a determination is made as to whether all of the data in the sector is in a non-programmable state and/or the sector-associated ECC indicates an error and based upon the outcome, a determination is made as to whether or not the end-of-file is reached.

[0031] Additionally, when there are more than one power failures in a media, there may be more than one corrupted sector and thus the system must verify all locations and where new file locations are started and ended. In other words, the system can maintain a table of corrupted files in a spare area so that after power-up, the system can identify and map the corrupted files thereby preventing their use for future storage.

[0032] Illustratively, in Fig. 6, the start-of-file is designated at sector 0 at 102 and remaining sectors of the file are stored in following sector locations, such as sector 1 104 and sector 2 106. However the next sector location, sector 3 108 has been adversely affected or corrupted by a power failure, thus, a new file will be stored starting from 110.

[0033] In summary, the embodiments of the invention relate to one time programmable non-volatile memories used for data such as camera images to allow data to be updated even though the memory medium cannot be overwritten. The embodiments provide one-time programmable memory for writing files in applications requiring re-writing of sector information, such as digital cameras, on a partial programmable basis, i.e. bit-by-bit, byte-by-byte or word-by-word basis, while maintaining backward compatibility with DOS-like or other known file systems.

[0034] The above is achieved by comparing a sector that is being re-written by the host to the same sector that was previously written by the host and by identifying, on a bit-by-bit, byte-by-byte or word-by-word basis, which bits, bytes or words of the previously written sector have been modified since, typically, not all of the bits of a sector are modified each time it is re-written by the host. Once this identification has been made, one of the three methods can be performed.

[0035] One method is to determine if the bits, bytes or words being modified, or identified to be modified, have even been programmed the first time around in the one-time programmable memory. In better understanding this, it should be recognised that even a write of a sector for the first time may not necessarily program all of the bits from a binary state of '1' to a binary state of '0'. A binary state of '1' in non-volatile memory, while meaning '1' or a high state, is nevertheless a non-programmed state because after initial manufacturing of the memory, all bits are programmed to '1'.

[0036] If it is determined that the bit, byte or word being re-written was not even programmed the previous time the sector was written, then it is programmed during the re-write and if it is determined the modified bit, byte or word was programmed the last time the sector was written, then the modified information is written to a spare area in a location that is within the same sector that is being re-written and in yet a third method, the modified information is written to a spare location outside of the sector being re-written, such as the spare locations discussed in the specification. In the last two

methods, a pointer is used to point to the spare area that now contains the modified bits, bytes or words for future reading of the re-written sector. This pointer is kept in the overhead portion of the sector in the one-time programmable memory. For a sector that is to be re-written, the pointer maintains track of the bits, bytes or words that are changed but it does so for the sector that it is presently a part thereof. For example, taking sector X as an example, initially, sector X is stored along with its overhead in a particular location in the one-time programmable memory and if it is re-written by the host, the new version of the sector, along with its overhead, is written to a different area within the one-time memory with the pointer in the different area being a part of the overhead but reflecting the changed bits, bytes or words. Thus, the same area or location within the one-time memory is not occupied by the pointer reflecting a re-write of the sector, rather, a different location within the one-time memory is used to store the pointer reflecting the re-write.

[0037] An alternative method is to simply write the modified bit, byte or words in the spare area altogether and to use a pointer to point to the spare location for future reading of the re-written sector.

[0038] The operation of the controller device 54 is under the control of micro-processor 74 which operates in accordance with instructions defined by a computer program.

[0039] An aspect of the present invention thus provides a storage medium storing processor implementable instructions for controlling a processor to carry out the method as hereinabove described.

[0040] Further, the computer program can be obtained in electronic form for example by downloading the code over a network such as the internet. Thus in accordance with another aspect of the present invention there is provided an electrical signal carrying processor implementable instructions for controlling a processor to carry out the method as hereinbefore described.

[0041] Although the present invention has been described in terms of specific embodiments it is anticipated that alterations and modifications thereof will no doubt become apparent to those skilled in the art. It is therefore intended that the following claims be interpreted as covering all such alterations and modification as fall within the scope of the invention.

Claims

1. A digital equipment system comprising:

   a. a host for sending commands to read or write files having sectors of information, each sector being modifiable on a bit-by-bit, byte-by-byte or word-by-word basis, said host being operative to receive responses to said commands;
   b. a controller device responsive to said commands, and including,

      one-time-programmable nonvolatile memory for storing information organized into sectors based on commands received from the host and upon receiving commands from the host to re-write a sector, the controller device being operable for re-writing said sector on a bit-by-bit, byte-by-byte or word-for-word basis.

2. A digital equipment system as recited in claim 1 wherein said one-time-programmable nonvolatile memory includes a system area and a data area, said system area storing information pertaining to the organization of the information stored or to be stored in the data area.

3. A digital equipment system as recited in claim 2 wherein said system area includes storage areas for including Original Engineering Manufacturer (OEM) identification/Bidirectional Input/Output System (BIOS), a File Allocation Table (FAT) 1, a FAT 2 and root disk directory information.

4. A digital equipment system as recited in claim 1 wherein said controller device further includes a first buffer for storing a host-provided sector and a second buffer for storing sectors stored or to be stored in the one-time-programmable nonvolatile memory.

5. A digital equipment system as recited in claim 4 further including a comparator coupled between said first and second buffer for comparing a sector to be modified or accessed by the host with those sectors to which information has been previously written.

6. A digital equipment system as recited in claim 1 wherein during power-up, said controller device is operable for identifying the end-of-file, wherein the location following the location in which the end-of-file resides is identified as the location for the start-of-file of a new file to be stored.

7. A digital equipment system as recited in claim 6 wherein said end-of-file is identified by the use of a flag.

8. A digital equipment system as recited in claim 1 wherein said files are digital photographs.

9. A digital equipment system as recited in claim 1 wherein said files are archives.

10. A digital equipment system as recited in claim 1 wherein said controller device maintains a correlation between logical addresses and physical addresses for translating host-provided addresses to addresses recognized by the one-time-program-

mable nonvolatile memory.

11. A digital equipment system as recited in claim 10 wherein said controller is operable for maintaining track of defective locations within the one-time-programmable nonvolatile memory.

12. A digital equipment system comprising:

a. a host for sending commands to read or write files having sectors of information, said host being operative to receive responses to said commands;
b. a controller device responsive to said commands, and including,

one-time-programmable nonvolatile memory for storing information organized into sectors, based on commands received from the host, and upon commands from the host to re-write a sector, said one-time-programmable nonvolatile memory including a spare area, said controller being operable for mapping sectors being re-written to spare area.

13. A digital equipment system as recited in claim 12 wherein said one-time-programmable nonvolatile memory further includes a system area and a data area.

14. A digital equipment system as recited in claim 13 wherein said controller device is operable for identifying a start-of-file location and an end-of-file location and a defective sector location within the one-time-programmable nonvolatile memory, the latter of which is skipped over when writing sectors.

15. A digital equipment system as recited in claim 14 wherein the information that was to be written to the defective sector is instead written to the spare area location.

16. A digital equipment system as recited in claim 12 wherein said controller is operable for determining if there is no start-of-file identifier at a location following a corrupted sector or there is no end-of-file in the rest of the one-time-programmable nonvolatile memory, such location identified as a corrupted sector due to power failure and designated accordingly so as to prevent future storage of information therein.

17. A digital equipment system comprising:

a. a host for sending commands to read or write files, said host being operative to receive responses to said commands;
b. a controller device responsive to said commands, and including,

one-time-programmable nonvolatile memory for storing files and identifying the start-of-file and end-of-file for a file being stored within the one-time-programmable memory, wherein during power-up, said controller device for identifying the end-of-file of a stored file, the location following the location in which the end-of-file resides being identified as the location for the start-of-file of a new file to be stored.

18. A digital equipment system comprising:

a. a host for sending commands to read or write files having sectors of information, said host being operative to receive responses to said commands;
b. a controller device responsive to said commands, and including,
one-time-programmable nonvolatile memory having spare locations for storing sector information, said one-time-programmable nonvolatile memory for storing information organized into sectors based on commands received from the host and upon receiving a command from the host to re-write or update a sector, the controller device for writing the updated sector to a spare location.

19. A digital equipment system comprising:

a. a host for sending commands to read or write files having sectors of information, each sector having associated therewith an error correction code (ECC) indicative of the corruption of sector information, said host being operative to receive responses to said commands;
b. a controller device responsive to said commands, and including,

one-time-programmable nonvolatile memory for storing information organized into sectors, wherein said controller checks the ECC of a particular sector for a determination of whether or not the particular sector is corrupted and if so, reads the information stored within the next sector and determines if the next sector information is in a non-programmable state and if so or the ECC associated with the next sector indicates that the next sector information is corrupt, the controller device identifies an end-of-file.

20. A digital equipment system comprising:

a. a host for sending commands to read or write files in portions corresponding to memory sectors, each sector being identifiable on a bit-by-bit, byte-by-byte or word-by-word basis, said

host being operable for commanding to re-write a previously written sector; and

b. a controller device responsive to the host commands and including, one-time-programmable non-volatile memory for storing information organized into sectors based on commands received from the host, said memory having stored therein the previously written sector and upon receiving a command from the host to re-write the previously written sector, the controller device being operable for performing partial programming of the previously written sector to write the sector to be re-written on a bit-by-bit, byte-by-byte or word-by-word basis.

21. A digital equipment system, as recited in claim 20, wherein the controller device is operable for comparing the sector to be re-written to the previously written sector and for identifying which bits, bytes or words of the sector to be re-written are modified and, if said identified bits, bytes or words were not previously programmed in the previously written sector, programming the same.

22. A digital equipment system, as recited in claim 20, wherein said memory includes spare locations and further wherein the controller device is operable for comparing the sector to be re-written to the previously written sector and for identifying which bits, bytes or words of the sector to be re-written are modified and writing said identified bits, bytes or words in said spare locations.

23. A method of re-writing sectors to previously written sectors within a one-time-programmable memory comprising:

a. receiving a command to re-write a previously written sector to a one-time-programmable memory, said previously written sector having bits, bytes or words;

b. comparing a sector to be re-written to said previously written sector for modifications to the latter, said sector to be re-written having bits, bytes or words;

c. identifying bits, bytes or words of the previously written sector that are modified;

d. determining if the modified bits, bytes or words of the previously written sector have not programmed; and

e. programming the determined modified bits, bytes or words of the previously written sector to make the previously written sector include the same information as the sector to be re-written.

24. A method of re-writing sectors to previously written sectors within a one-time-programmable memory comprising:

a. receiving a command to re-write a previously written sector to a one-time-programmable memory, said previously written sector having bits, bytes or words;

b. comparing a sector to be re-written to said previously written sector for modifications to the latter, said sector to be re-written having bits, bytes or words;

c. identifying bits, bytes or words of the previously written sector that are modified;

d. determining if the modified bits, bytes or words of the previously written sector have not programmed; and

e. writing the modified bits, bytes or words into a spare location within the one-time-programmable memory.

25. A memory system comprising:

an interface for receiving commands from a host for the writing of data to and reading of data from the memory system;

a one-time programmable memory having a data area divided into physical sectors;

a control device responsive to the commands and comprising means for emulating a re-programmable memory whereby logical addresses available to the host are presented to the host as being mapped to re-writable virtual sectors in the emulated memory and are actually mapped to physical addresses in the physical sectors in the one-time-programmable memory;

the control device comprising a comparator responsive to a command to rewrite a virtual sector with new data by comparing the new data with contents of a physical sector in which existing data for the virtual sector was previously programmed to identify portions of said new data which are different from the existing data; and

partial programming means for programming only the identified portions of the new data into previously unprogrammed areas of the memory and for maintaining mapping of the logical addresses for the re-written virtual sector to locations in the memory containing the new data.

26. A method of operating a memory system comprising a one-time programmable memory having a data area divided into physical sectors;:

the method comprising:

receiving commands from a host for the

writing of data to and reading of data from the memory system;

operating a control device responsive to the commands to emulate a re-programmable memory whereby logical addresses available to the host are presented to the host as being mapped to re-writable virtual sectors in the emulated memory and are actually mapped to physical addresses in the physical sectors in the one-time-programmable memory;

the control device being responsive to a command to rewrite a virtual sector with new data by comparing the new data with contents of a physical sector in which existing data for the virtual sector was previously programmed to identify portions of said new data which are different from the existing data; and

partial programming the memory by programming only the identified portions of the new data into previously unprogrammed areas of the memory and maintaining mapping of the logical addresses for the re-written virtual sector to locations in the memory containing the new data.

27. A storage medium storing processor implementable instructions for controlling a processor to carry out the method of any one of claims 23, 24, and 26.

28. An electrical signal carrying processor implementable instructions for controlling a processor to carry out the method of any one of claims 23, 24 and 26.

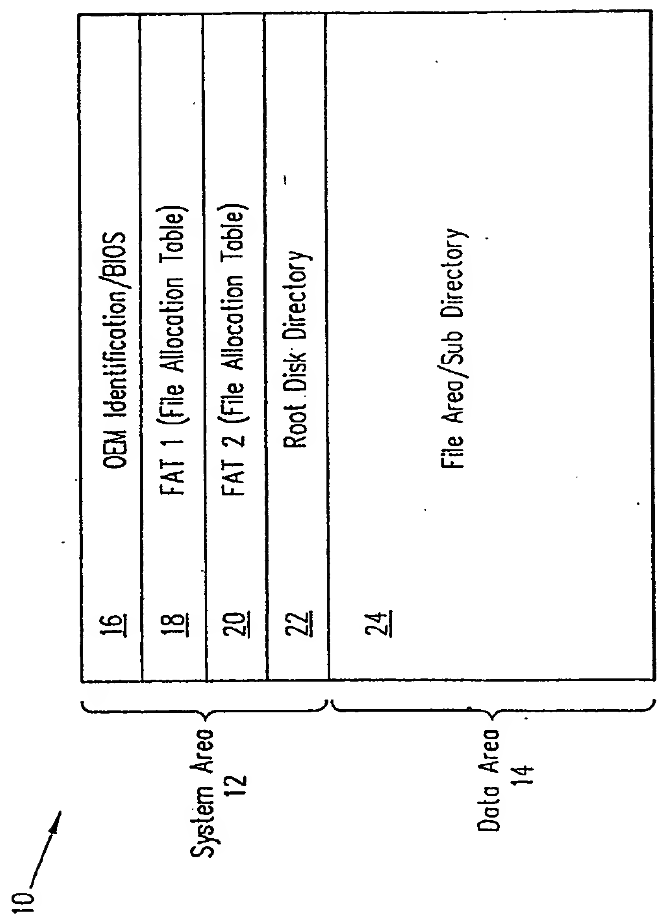
10
16 OEM Identification/BIOS
18 FAT 1 (File Allocation Table)
20 FAT 2 (File Allocation Table)
22 Root Disk Directory
24 File Area/Sub Directory
System Area 12
Data Area 14

FIG. 1

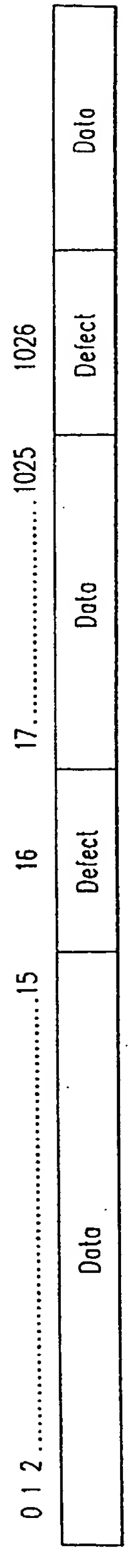
0 1 2 15
16
17 1025
1026
Data
Defect
Data
Defect
Data

FIG. 4

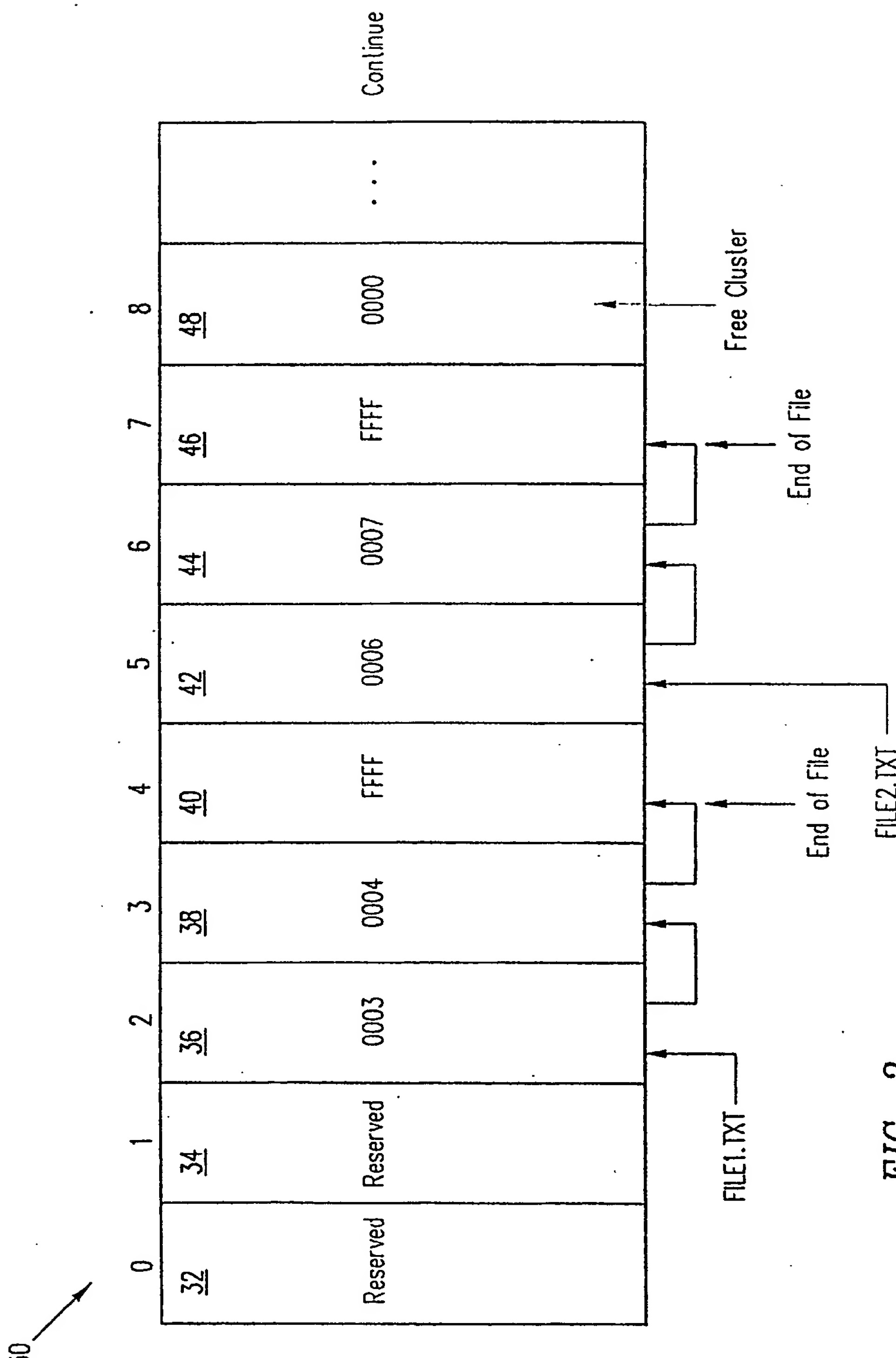
30
0
1
2
3
4
5
6
7
8
32
34
36
38
40
42
44
46
48
Reserved
Reserved
0003
0004
FFFF
0006
0007
FFFF
0000
. . .
Continue
FILE1.TXT
End of File
FILE2.TXT
End of File
Free Cluster

FIG. 2

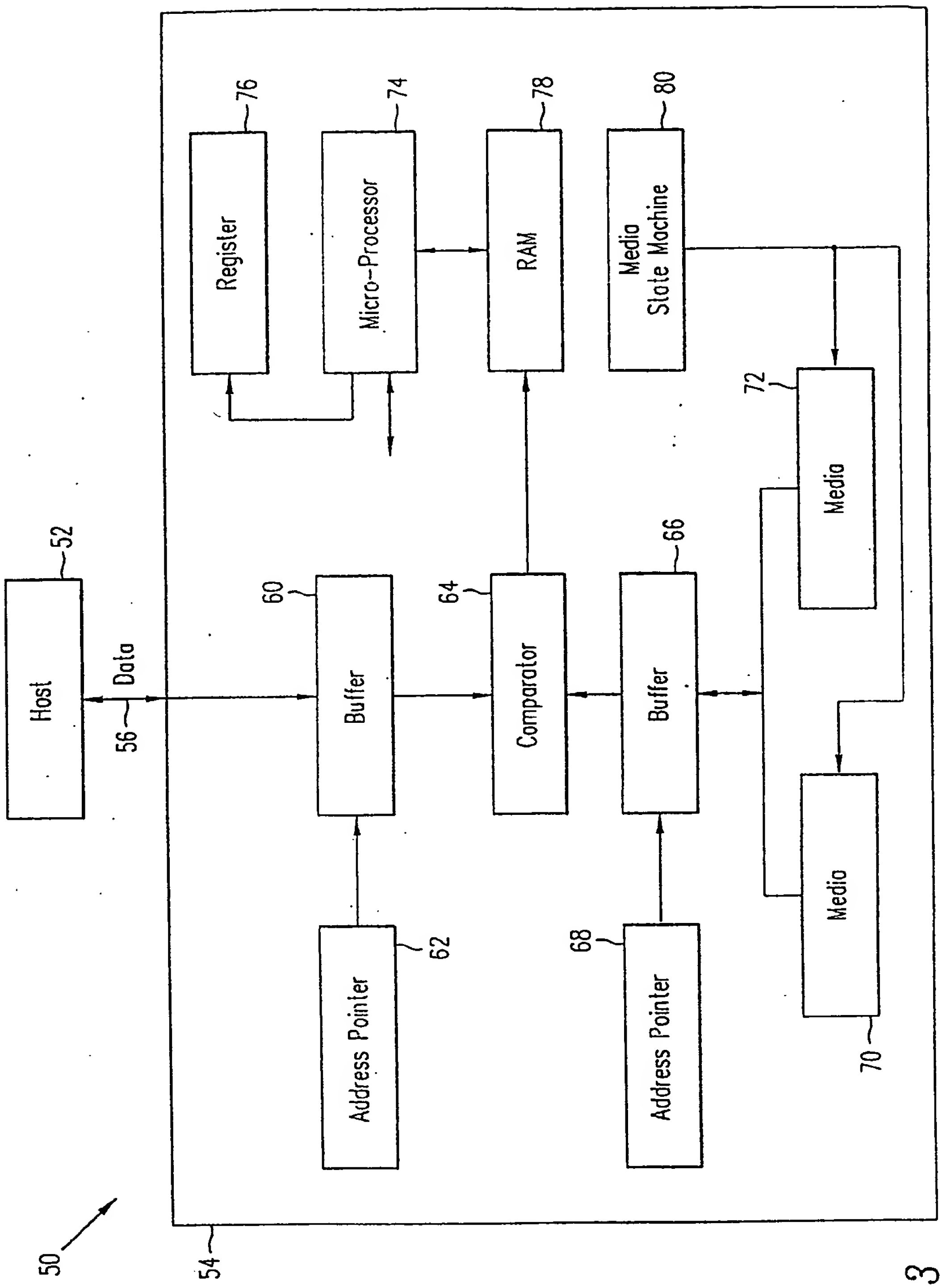
Host
52
Data
56
Register
76
Micro-Processor
74
RAM
78
Media
State Machine
80
Buffer
60
Comparator
64
Buffer
66
Media
72
Media
70
Address Pointer
62
Address Pointer
68
50
54

FIG. 3

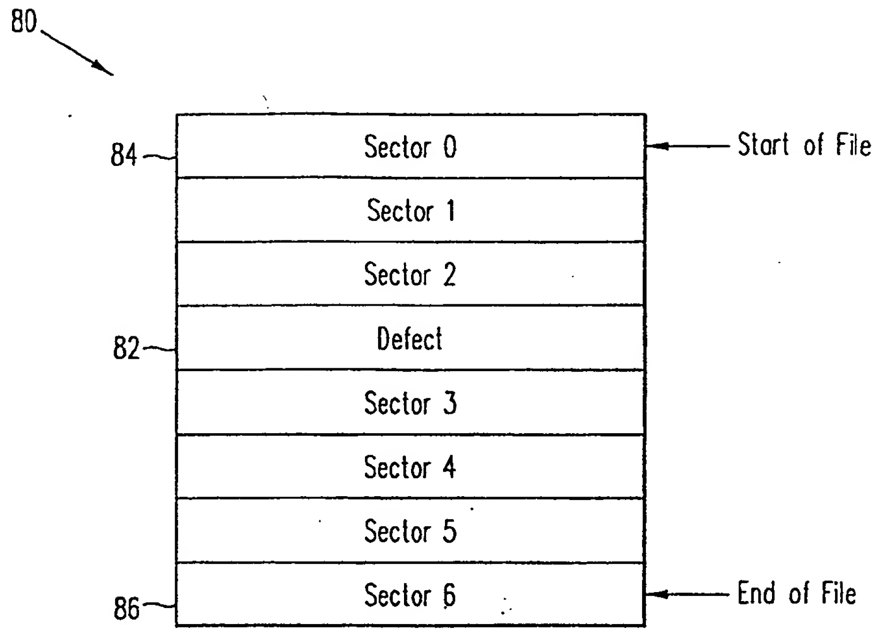
80
84
82
86
Sector 0
Sector 1
Sector 2
Defect
Sector 3
Sector 4
Sector 5
Sector 6
Start of File
End of File

FIG. 5

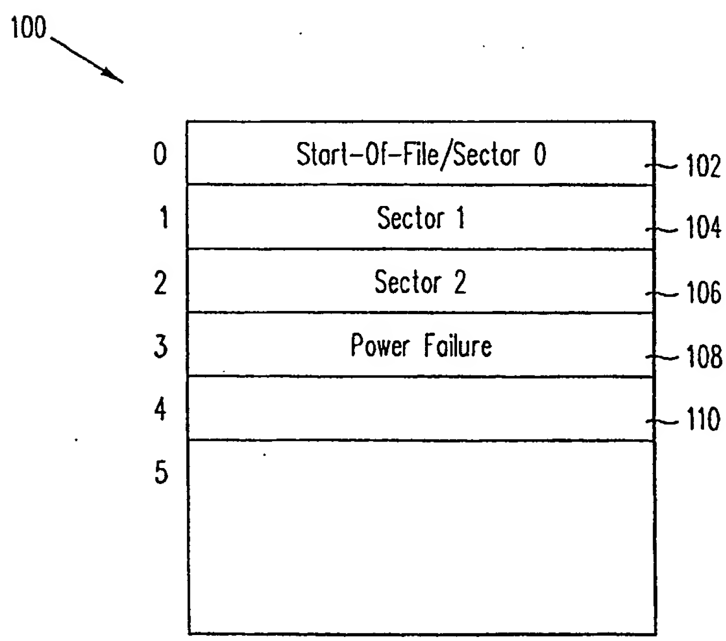
100
0
1
2
3
4
5
Start-Of-File/Sector 0
Sector 1
Sector 2
Power Failure
102
104
106
108
110

FIG. 6